

Serial No. 10/711,826  
Group Art Unit 2168  
Docket No: SVL920040022US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPEAL BRIEF – 37 C.F.R § 1.192

U.S. Patent Application 10/711,826 entitled:

“Method of Changing the Page Size of a DB2 Table Space While Keeping the Object Available”

**Real Party in Interest:** International Business Machines Corporation

**Related Appeals and Interferences:**

None

**Status of Claims:**

Claims 1-22 are pending.

Claims 1-22 are rejected.

Claims 1, 3, 4, and 9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Subramaniam et al. (6,965,899) in view of Teng et al. (6,460,048).

Claims 2, 5-8, and 10-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Subramaniam et al. (6,965,899) in view of Teng et al. (6,460,048) and further in view of Huras et al. (2001/0047360).

Claims 13-14 and 18-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Iyer et al. (2002/0143743) in view of Sockut et al. (Database Reorganization – Principles and Practice).

Claims 15-17 and 20-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Iyer et al. (2002/0143743) in view of Sockut et al. (Database Reorganization – Principles and Practice) and further in view of Teng et al. (6,460,048).

**Status of Amendments:**

No Amendments were presented after the Final Office Action of 02/23/2007.

**Summary of Claimed Subject Matter:**

(NOTE: All citations are made from the original specification, including the figures.)

The present invention, according to independent **claim 1**, provides for a method (*see figure 2 in the application-as-filed*) for updating object page size during reorganization of a table space in a database comprising the steps of: allocating a shadow data set for at least one object belonging to a first data set from said table space (*see paragraph [0019] and step 200 of figure 2 in the application-as-filed*); writing to a shadow control block corresponding to each of said allocated shadow data sets, a page size value larger than a page size value to be allocated; said larger page size value corresponding to said at least one object (*see paragraph [0019] and step 204 of figure 2 in the application-as-filed*); loading rows from said first data set of said table space into said allocated shadow data set; for each row loaded, reading each object corresponding to said loaded row from said table space and writing said read object to said allocated shadow data set (*see paragraph [0019] and step 206 of figure 2 in the application-as-filed*); and updating at least: said first data set of said table space with data from said shadow data set; a system catalog for said database with said larger page size value; and at least one database control block with said larger page size value; said at least one database control block corresponding to said first data set (*see paragraph [0019] and step 210 of figure 2 in the application-as-filed*).

The present invention, according to independent **claim 9**, provides for an article of manufacture comprising a computer usable medium having computer readable program code

embodied therein which implements a method (*see figure 2 in the application-as-filed*) for updating object page size during reorganization of a table space in a database, said medium comprising modules implementing: allocating a shadow data set for at least one object belonging to a first data set from said table space (*see paragraphs [0019], [0020], [0021], [0022] and step 200 of figure 2 in the application-as-filed*); writing to a shadow control block corresponding to each of said allocated shadow data sets, a page size value larger than a page size value to be allocated; said larger page size value corresponding to said at least one object (*see paragraphs [0019], [0020], [0021], [0022] and step 204 of figure 2 in the application-as-filed*); loading rows from said first data set of said table space into said allocated shadow data set; for each row loaded, reading each object corresponding to said loaded row from said table space and writing said read object to said allocated shadow data set (*see paragraphs [0019], [0020], [0021], [0022] and step 206 of figure 2 in the application-as-filed*); and updating at least: said first data set of said table space with data from said shadow data set; a system catalog for said database with said larger page size value; and at least one database control block with said larger page size value; said at least one database control block corresponding to said first data set (*see paragraphs [0019], [0020], [0021], [0022] and step 210 of figure 2 in the application-as-filed*).

The present invention, according to independent **claim 13**, provides for a method for reorganizing a designated object of a database that has exceeded a current page size by: writing to a larger page, rows added to said designated object (*see paragraphs [0009], [0010] and step 204 of figure 2 in the application-as-filed*); permitting continual access to said designated object during said writing step (*see paragraphs [0009] and [0010] in the application-as-filed*); reading

constituent rows from a plurality of existing pages corresponding to said designated object and subsequently copying said constituent rows to said larger page (*see paragraphs [0009] and [0010] in the application-as-filed*); and externalizing said designated object (*see paragraphs [0009] and [0010] in the application-as-filed*).

The present invention, according to independent **claim 18**, provides an article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements the reorganization of a designated object of a database that has exceeded a current page size; said medium comprising modules implementing: writing to a larger page, rows added to said designated object (*see paragraphs [0009], [0010] and step 204 of figure 2 in the application-as-filed*); permitting continual access to said designated object during said writing step (*see paragraphs [0009] and [0010] in the application-as-filed*); reading constituent rows from a plurality of existing pages corresponding to said designated object and subsequently copying said constituent rows to said larger page (*see paragraphs [0009] and [0010] in the application-as-filed*); and externalizing said designated object (*see paragraphs [0009] and [0010] in the application-as-filed*).

**Grounds of Rejection to be Reviewed on Appeal:**

1. Claims 1, 3, 4, and 9 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. 6,965,899 (Subramanian et al.) in view of U.S. 6,460,048 (Teng et al.). Claims 2, 5-8, and 10-12 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. 6,965,899 (Subramanian et al.) in view of U.S. 6,460,048 (Teng et al.) and further in view of U.S. Pub. 2001/0047360 (Huras et al.). Claims 13-14, and 18-19 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Pub. 2002/0143743 (Iyer et al.) in view of “*Database Reorganization – Principles and Practice*” (Socket et al.). Claims 15-17, and 20-22 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Pub. 2002/0143743 (Iyer et al.) in view of “*Database Reorganization – Principles and Practice*” (Socket et al.) and further in view of U.S. 6,460,048 (Teng et al.). **With respect to pending claims 1-22, was a proper rejection made under 35 U.S. C. § 103(a) using existing USPTO guidelines?**

**ARGUMENT:**

**1. With respect to pending claims 1-22, was a proper rejection made under 35 U.S. C. § 103(a) using existing USPTO guidelines?**

Claims 1, 3, 4, and 9 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. 6,965,899 (Subramanian et al.) in view of U.S. 6,460,048 (Teng et al.). Claims 2, 5-8, and 10-12 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. 6,965,899 (Subramanian et al.) in view of U.S. 6,460,048 (Teng et al.) and further in view of U.S. Pub. 2001/0047360 (Huras et al.). Claims 13-14, and 18-19 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Pub. 2002/0143743 (Iyer et al.) in view of “*Database Reorganization – Principles and Practice*” (Socket et al.). Claims 15-17, and 20-22 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Pub. 2002/0143743 (Iyer et al.) in view of “*Database Reorganization – Principles and Practice*” (Socket et al.) and further in view of U.S. 6,460,048 (Teng et al.).

To establish a prima facie case of obviousness under U.S.C. § 103, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. Additionally, the teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure (In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir.

1991))).

Applicants contend that the above-mentioned specific combinations of references fail to provide many of the features of the Applicants' claims.

With respect to independent claims 1 and 9, the Examiner contends that the Subramaniam and Teng references, in combination, teach all the features of Applicants' claims 1 and 9. Applicants respectfully disagree with this assertion.

Applicants' claim 1 provides for a method for updating object page size during reorganization of a table space in a database comprising the steps of: allocating a **shadow data set** for at least one object belonging to a first data set from said table space; writing to a **shadow control block** corresponding to each of said allocated shadow data sets, a **page size value larger than a page size value to be allocated**; said larger page size value corresponding to said at least one object; loading rows from said first data set of said table space into said allocated shadow data set; for each row loaded, reading each object corresponding to said loaded row from said table space and writing said read object to said allocated shadow data set; and updating **at least**: said first data set of said table space with data from said shadow data set; a system catalog for said database with said larger page size value; **and** at least one database control block with said larger page size value; said at least one database control block corresponding to said first data set.



Applicants claim 9 provides for an article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements the method as substantially described in claim 1.

Subramaniam et al., by contrast, teaches a method for modifying a target table within a relational database. Subramaniam's method comprises the steps of: creating a revised table that has one or more attributes that are different than corresponding attributes of the target table. While the revised table is being created, users are prevented from accessing the revised table, but are allowed to access the target table. According to Subramaniam's method, when creation of the revised table is complete, the target table is locked and the revised table is synchronized with the target table, causing all subsequent attempts to access the target table to access the revised table.

Teng et al., also by contrast, teaches a method for reorganizing a database object, wherein the database object is comprised of at least one database file. Teng et al.'s method comprises the steps of: providing source database files including data for the database objects subject to a reorganization, wherein the source database files have source names; creating shadow copies of the source database files; generating shadow names for the shadow copies, wherein the source names and corresponding shadow names are different; and reorganizing data in the shadow copies including database objects, wherein after the reorganization, the shadow names are used to access the database files for the reorganized database objects.

Independent claims 1 and 9 of the present invention specifically teach a method for **updating object page size during reorganization of a table space**. Specifically, as mentioned above, both claims 1 and 9 teach writing, to a shadow control block, a page size value larger than a page size value to be allocated. With respect to this feature, the Examiner asserts that column 2, line 59 through column 3, line 12 of Subramaniam teaches such a feature. Column 2, line 59 through column 3, line 12 is reproduced below:

“The online modification technique allows the database administrator to perform various types of modifications to the target table, including but not limited to: 1) modify the storage parameters of the target table, 2) add support for parallel queries to be performed on the target table after the target table is modified, 3) change a heap organized target table to an Index Only Table (IOT) or vice versa, 4) move the target table to a different tablespace, 5) add or drop partitioning support for the target table, 6) recreate the target table in order to reduce fragmentation, 7) add, drop, or rename columns in the target table, 8) transform data in the target table.

FIG. 1 is a flowchart that illustrates the online modification operation of a target table in a relational database. At block 102 of FIG. 1, an empty revised table is created with all the attributes that are desired for the online modification. At block 104, the database

system starts populating the revised table based on data in the target table. Data from the target table is copied over to the revised table based on column mapping information. Column mapping information is explained in greater detail herein.” (emphasis added).

By Subramaniam’s own admission, the Examiner’s citation merely teaches an “online modification technique” that allows the database administrator to perform various types of modifications to a target table. **Absent from the above-reproduced Examiner’s citation and absent from the entire Subramaniam reference is ANY mention of page size.** It appears that the Examiner is erroneously equating the size of a target table described in the citation to Applicants’ page size. It should be noted that parameters such as **page size** are specific parameters in the database arena and **CANNOT** be equated to a size of a table. Applicants submit that since Subramaniam fails to mention page size in its entire specification, it would erroneous to argue that Subramaniam teaches the step of writing to a shadow control block a page size value larger than a page size value to be allocated.

The Examiner was given an opportunity to specifically address the failure of the Subramaniam reference to mention page size and the failure of the Subramaniam reference to teach the step of writing, to a shadow control block, a page size value larger than a page size value to be allocated. However, the Examiner once again has chosen to reiterate the same arguments presented in the previous office action and has added on page 13 of the Office Action

that “**adding columns and revising a table**” would teach such a feature.

Once again, it appears that the Examiner is relying on an **erroneous assumption that the size of the table and page size are equivalent parameters**. Applicants respectfully submit that this assumption on the part of Examiner is **erroneous**, as these are separate parameters that cannot be equated together. For example, page size in database technology refers to the way database entries are stored in a database, wherein the page size are allocated as 512 bytes or 64k bytes. Applicants’ claims 1 and 9 teach writing, to a shadow control block, a page value that is larger than such an allocated value (i.e., if 512 bytes is the allocated value, the method of claim 1 and the article of manufacture of claim 9 teach writing to a show control block a value larger than the allocated 512 bytes). Such a feature is neither taught not suggested by the Subramaniam reference or the Teng reference.

Applicants agree with the Examiner’s conclusion on page 3 of the Office Action of 02/23/2007 that Subramaniam reference does NOT teach claim 1 and 9’s feature of “**updating at least: said first data set of said table space with data from said shadow data set; a system catalog for said database with said larger page size value; and at least one database control block with said larger page size value; said at least one database control block corresponding to said first data set**”. However, Applicants respectfully disagree with the Examiner that such a feature is remedied by the Teng reference.

Specifically, for support, the Examiner cites column 6, lines 55-65 of Teng as teaching

this feature of claims 1 and 9. Column 6, lines 55-65 of Teng merely reference Figure 2 which teaches the reorganization of “shadow copies of data sets”. However, Applicants respectfully assert that neither Teng’s Figure 2 nor Teng’s description of Figure 2 attempts to teach or suggest **updating a system catalog for said database with said larger page size value**. Applicants also assert that Teng’s reorganization procedure shown in Figure 2 and accompanying description fail to teach or suggest **updating one database control block with said larger page size value**. Further, absent in the Examiner’s citations and the entire Teng reference is a teaching for **updating a first data set of the table space with data from a shadow set**. As **each** of the above-mentioned features are absent in Subramaniam and Teng references, it would be moot to argue that a **combined teaching for all three updates** can be found in the Subramaniam and Teng references (see, for example, claim 1 – “updating **at least**: said first data set of said table space with data from said shadow data set; a system catalog for said database with said larger page size value; **and** at least one database control block with said larger page size value”).

Applicants wish to note that the above-mentioned arguments for independent claims 1 and 9 substantially apply to dependent claims 2-8 and 10-12 as they inherit all the features of the claim from they depend.

Independent claims 13 and 18 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Pub. 2002/0143743 (Iyer et al.) in view of “*Database Reorganization – Principles and Practice*” (Socut et al.).

Claim 13 provides for a method for reorganizing a **designated object of a database that has exceeded a current page size** by: writing to a **larger page**, rows added to said designated object; permitting continual access to said designated object during said writing step; reading constituent rows from a plurality of existing pages corresponding to said designated object and subsequently copying said constituent rows to said **larger page**; and externalizing said designated object.

Applicants' claim 18 provides for an article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements the method as substantially described in claim 13.

While Applicants agree with the Examiner's conclusion that the feature of a larger page is not taught or suggested by the Iyer reference, Applicants respectfully disagree with the Examiner that such a feature is taught by page 386, column 2, second paragraph 2 of the Sockut reference.

Sockut merely provides a general overview of database reorganization methods. On page 386, Sockut merely describes utilities that change page sizes by copying without using an intermediate unload file. Such changes are in line with what was described as prior art in the application-as-filed, and suffer from the same problems and performance degradation associated with fragmentation and badly clustered indices.

Absent in the Examiner's citation is a teaching for **writing to a larger page when designated object has exceeded a current page size**. It should be noted that just a mere mention of an utility that changes page size **CANNOT** be equated to writing to a larger page when a designated object has exceeded a current page size.

Further, absent in the citation of Sockut, or the entire Sockut reference, is a teaching or suggestion for **writing to a larger page, rows added to a designated object to be reorganized**. Also, absent in the Sockut reference is a teaching or suggestion for **permitting continual access to said designated object during said writing step**. Further absent in the Sockut reference is a teaching or suggestion for reading constituent rows from a plurality of existing pages corresponding to said designated object and subsequently **copying said constituent rows to said larger page** and **externalizing said designated object**. Absent such teachings, the Sockut reference, either by itself or in combination with the Iyer reference, cannot render obvious the teachings of claims 13 and 18.

Applicants wish to note that the above-mentioned arguments for independent claims 13 and 18 substantially apply to dependent claims 14-17 and 19-22 as they inherit all the features of the claim from they depend.

In addition, with respect to dependent claims 14 and 19, the Examiner states on page 10 of the Office Action of 02/23/2007 that the Iyer reference on Paragraph [0056] teaches the

feature of constituent rows of the designated object being re-arranged in physical storage to **eliminate fragmentation**.

Paragraph [0056] of Iyer is reproduced below:

“There are effects on storage structures during insertion of a row or during growth by update (modification) of a variable-length column of an existing row. During these operations, *if the desired page lacks enough contiguous free space* (the space available for insertions and growth), the DBMS 106 *compacts* the page to make its free space contiguous. *During compaction, when the DBMS 106 moves a record, the DBMS 106 updates the ID map's pointer to the record; compaction does not change the record's RID*. If *compaction produces enough free space*, the data goes into the desired page.” (emphasis added).

Paragraph [0056] of Iyer merely teaches **compaction wherein a database record is moved if the desired page lacks enough contiguous free space**. Such compaction **CANNOT** be equated to Applicants' claims 14's and 19's feature of constituent rows of the designated object being re-arranged in physical storage to **eliminate fragmentation**.

Also, with respect to claims 17 and 22, the Examiner on page 12 of the Office Action of 02/23/2007 states that the Teng reference discloses the feature of control information associated



with the **system catalog being updated to reflect a change in page size corresponding to said externalized designated object**. For support, the Examiner cites column 6, lines 55-65 of Teng as teaching such a feature. A portion of the Examiner's citation is provided below:

“....After completely updating the shadow copies with the log entries, the **database program 4 then updates (at block 70) system information, such as the IPREFIX field in the system tables or the DBD, to indicate that the current version of the fifth instance qualifier is the value of the fifth instance qualifier of the shadow copies of the data sets...**” (emphasis added).

Applicants respectfully assert (and as can be seen above) that column 6, lines 55-65 of Teng merely references updating “system information, such as the IPREFIX field in the system tables or the DBD **to indicate the current version**”. Such a feature of updating to indicate the current version **CANNOT** be equated to **updating to reflect a change in the page size** corresponding to the externalized designated object.

Hence, at least for the reasons set forth above, Applicants respectfully maintain that the above-mentioned specific combinations of references fail to provide many of the features of the Applicants' pending claims 1-22. Hence, Applicants respectfully assert that the Examiner has failed to establish a *prima facie* case of obviousness, and further assert that an improper 35 U.S.C. §103(a) rejection was issued with regards to claims 1-22.

SUMMARY

As has been detailed above, none of the references, cited or applied, provide for the specific claimed details of applicant's presently claimed invention, nor render them obvious. It is believed that this case is in condition for allowance and reconsideration thereof and early issuance is respectfully requested.

As this Appeal Brief has been timely filed within the set period of response, no fee for extension of time is required. However, the Commissioner is hereby authorized to charge any deficiencies in the fees provided, including extension of time, to Deposit Account No. 09-0460.

Respectfully submitted by  
Applicant's Representative,  
*/ramraj soundararajan/*

Ramraj Soundararajan  
Reg. No. 53,832

IP Authority, LLC.  
9435 Lorton Market Street  
#801  
Lorton, VA 22079  
(571)642-0033

June 27, 2007

**Claims Appendix:**

**1. (Previously Presented)** A method for updating object page size during reorganization of a table space in a database comprising the steps of:

- (a) allocating a shadow data set for at least one object belonging to a first data set from said table space;
- (b) writing to a shadow control block corresponding to each of said allocated shadow data sets, a page size value larger than a page size value to be allocated; said larger page size value corresponding to said at least one object;
- (c) loading rows from said first data set of said table space into said allocated shadow data set; for each row loaded, reading each object corresponding to said loaded row from said table space and writing said read object to said allocated shadow data set; and
- (d) updating at least: said first data set of said table space with data from said shadow data set; a system catalog for said database with said larger page size value; and at least one database control block with said larger page size value; said at least one database control block corresponding to said first data set.

**2. (Original)** A method for updating object page size during reorganizing a table space in a database, as per claim 1, wherein said method further comprising the steps of:

prior to said shadow data set allocation, blocking write access to

said first data set from said table space; and

subsequent to said updating said table space, said database system catalog, and said at least one database control block, allowing write operations related to said first data set to proceed.

**3. (Previously Presented)** A method for updating object page size during reorganizing a table space in a database, as per claim 1, wherein said method is implemented across network elements.

**4. (Previously Presented)** A method for updating object page size during reorganizing a table space in a database, as per claim 3, wherein said across network elements is any of the following: local area network (LAN), wide area network (WAN), or the Internet.

**5. (Original)** A method for updating object page size during reorganizing a table space in a database, as per claim 1, wherein said loading is further comprised of:

(a) concurrently loading rows corresponding to said at least one object from said table space into said shadow data set and extracting index keys for each loaded row; said shadow data set allocated for each of said at least one object and associated indices, and

(i) for each of said loaded rows, identifying columns representing data corresponding to said at least one object; and

(ii) for each column representing data corresponding to said at least one object, reading data from said table space; said data read

using row information from a currently loaded row; and writing said data  
corresponding to said at least one object to said shadow data set.

**6. (Original)** A method for updating object page size during reorganizing a table space in a database, as per claim 5, wherein said method further comprising the steps of:

prior to said concurrent loading of rows and extracting of index keys, unloading  
rows from said table space; and  
sorting said unloaded rows; said sorted rows subsequently loaded into said  
shadow data set in said loading step.

**7. (Original)** A method for updating object page size during reorganizing a table space in a database, as per claim 5, wherein said method is implemented across network elements.

**8. (Original)** A method for updating object page size during reorganizing a table space in a database, as per claim 7, wherein said across network elements is any of the following: local area network (LAN), wide area network (WAN), or the Internet.

**9. (Original)** An article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements a method for updating object page size during reorganization of a table space in a database, said medium comprising modules implementing:

(a) allocating a shadow data set for at least one object belonging to a first data set  
from said table space;

- (b) writing to a shadow control block corresponding to each of said allocated shadow data sets, a page size value larger than a page size value to be allocated; said larger page size value corresponding to said at least one object;
- (c) loading rows from said first data set of said table space into said allocated shadow data set; for each row loaded, reading each object corresponding to said loaded row from said table space and writing said read object to said allocated shadow data set; and
- (d) updating at least: said first data set of said table space with data from said shadow data set; a system catalog for said database with said larger page size value; and at least one database control block with said larger page size value; said at least one database control block corresponding to said first data set.

**10. (Original)** An article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements a method for updating object page size during reorganization of a table space in a database, as per claim 9, wherein:

prior to said shadow data set allocation, blocking write access to said first data set from said table space; and

subsequent to said updating said table space, said database system catalog, and said at least one database control block, allowing write operations related to said first data set to proceed.

**11. (Original)** An article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements a method for updating object page size during reorganization of a table space in a database, as per claim 9, wherein said loading is further comprised of:

(a) concurrently loading rows corresponding to said at least one object from said table space into said shadow data set and extracting index keys for each loaded row; said shadow data set allocated for each of said at least one object and associated indices, and

(i) for each of said loaded rows, identifying columns representing data corresponding to said at least one object; and

(ii) for each column representing data corresponding to said at least one object, reading data from said table space; said data read using row information from a currently loaded row; and writing said data corresponding to said at least one object to said shadow data set.

**12. (Original)** An article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements a method for updating object page size during reorganization of a table space in a database, as per claim . A method for updating object page size during reorganizing a table space in a database, as per claim 11, wherein said method further comprising the steps of:

prior to said concurrent loading of rows and extracting of index keys, unloading

rows from said table space; and  
sorting said unloaded rows; said sorted rows subsequently loaded into said  
shadow data set in said loading step.

**13. (Previously Presented)** Reorganizing a designated object of a database that has exceeded a current page size by:

- (a) writing to a larger page, rows added to said designated object;
- (b) permitting continual access to said designated object during said writing step;
- (c) reading constituent rows from a plurality of existing pages corresponding to  
said designated object and subsequently copying said constituent rows to  
said larger page; and
- (d) externalizing said designated object.

**14. (Previously Presented)** Reorganizing a designated object of a database that has exceeded a current page size, as per claim 13, wherein during said copying, constituent rows of said designated object are re-arranged in physical storage to eliminate fragmentation.

**15. (Previously Presented)** Reorganizing a designated object of a database that has exceeded a current page size, as per claim 13, wherein said database is comprised of: a plurality of index values and a system catalog.

**16. (Previously Presented)** Reorganizing a designated object of a database that has exceeded a current page size, as per claim 15, wherein during said copying, data in said constituent rows is



compacted and is stored, on contiguous pages in physical storage, in accordance with one of said plurality of index values.

**17. (Previously Presented)** Reorganizing a designated object of a database that has exceeded a current page size, as per claim 15, wherein control information associated with said system catalog is updated to reflect a change in page size corresponding to said externalized designated object.

**18. (Previously Presented)** An article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements the reorganization of a designated object of a database that has exceeded a current page size; said medium comprising modules implementing:

- a) writing to a larger page, rows added to said designated object;
- b) permitting continual access to said designated object during said writing step;
- c) reading constituent rows from a plurality of existing pages corresponding to said designated object and subsequently copying said constituent rows to said larger page; and
- d) externalizing said designated object.

**19. (Previously Presented)** An article of manufacture comprising a computer usable medium, as per claim 18, wherein during said copying, constituent rows of said designated object are re-arranged in physical storage to eliminate fragmentation.

**20. (Previously Presented)** An article of manufacture comprising a computer usable medium, as per claim 18, wherein said database is comprised of: a plurality of index values and a system catalog.

**21. (Previously Presented)** An article of manufacture comprising a computer usable medium, as per claim 20, wherein during said copying, data in said constituent rows is compacted and is stored, on contiguous pages in physical storage, in accordance with one of said plurality of index values.

**22. (Previously Presented)** An article of manufacture comprising a computer usable medium, as per claim 20, wherein control information associated with said system catalog is updated to reflect a change in page size corresponding to said externalized designated object.

Serial No. 10/711,826  
Group Art Unit 2168  
Docket No: SVL920040022US1

## **Evidence Appendix**

None

Serial No. 10/711,826  
Group Art Unit 2168  
Docket No: SVL920040022US1

### **Related Proceedings Appendix**

None